

Biclustering based Collaborative Filtering Algorithm for Personalized Web Service Recommendation

M. Chandralekha

PG Student

Department of CSE

PSG College of Technology

Peelamedu, Coimbatore,

641004, India

Saranya K.G.

Assistant Professor (Sr.G)

Department of CSE

PSG College of Technology

Peelamedu, Coimbatore,

641004, India

G. Sudha Sadasivam,

PhD

Professor

Department of CSE

PSG College of Technology

Peelamedu, Coimbatore,

641004, India

ABSTRACT

Collaborative filtering (CF) is a technique to carry out automatic suggestions for a user based on the view of other users with similar taste. Most of the CF algorithms do not consider the existent duality between users and items, taking into account only the similarities between users or only the similarities between items. Though, there are some problems such as data sparsity which limit its further progress. To deal with the data sparsity problem a novel collaborative filtering recommendation algorithm is proposed based on biclustering. By taking into consideration the biclustering method to carry out clustering of rows and columns at the same time, the algorithm is able to group similarities between users and items. The paper also presents the comparison of user-based clustering and biclustering. In order to evaluate the proposed methodology, the Web Service (WSDL) dataset is applied to it which contains user's ratings to a large set of web services. The results indicate that the proposed methodology is able to provide useful recommendations for the users, especially in the presence of sparse data. Furthermore, the robustness of the proposed approach increases the data sparsity and the number of users that outperforms other methodologies for CF.

Keywords

Biclustering, Collaborative filtering, Recommendation system, Web service.

1. INTRODUCTION

When people prefer between a varieties of options with which they do not have any experience (e.g., cars, movies), they usually rely on suggestions from others who have such experience [16], [17]. Though, there are thousands or millions of options, like in the Web, it becomes impossible to find out an expert that can give suggestions for random options. As an attempt to handle with this problem, recommendation systems have been proposed as a way to help people to deal with information overload. Recommendations are created from rating information of more users with similar interests. The process, called Collaborative Filtering, is a technique that applies data mining and information retrieval techniques to carry out automated recommendations, for a user based upon the guess that if users have agreed in the past then they tend to agree in the future. Usually, a Collaborative Filtering algorithm takes input as a matrix K_{pq} where each entry represents the rating of user p for item q . Thus, the Collaborative Filtering algorithm correlates users based on their ratings.

There are two families of Collaborative Filtering algorithms; the first one is the nearest-neighbors family, also known as memory-based, which recommends based on the preferences

of the nearest-neighbors. The second one contains model-based algorithms, which recommend by initially developing a model of user ratings. Since similar research has reported that nearest-neighbor algorithms present improved performance in terms of accuracy when compared with model-based algorithms, this paper will focus in the previous approach.

With respect to nearest-neighbor algorithms, there exist two main approaches: the user-based, which construct neighborhoods depending on the similarity between users; and item-based, which construct neighborhoods depending on the similarities between items. Both user-based and item-based finds out the similarities either between users or between items respectively. So, they avoid the combined analysis of users and items. Additionally, user-based and item-based algorithms cannot disclose the incomplete matching of preferences, because their similarity measures include the whole set of items or users, respectively. Though, two users share similar preferences only for a subset of items, it is desirable that the Collaborative Filtering algorithm is capable of dealing with this scenario. An efficient way of taking into account the duality users-items into a Collaborative Filtering algorithm is by using the so called biclustering technique, which does clustering of rows and columns at the same time.

The proposed approach focuses on a novel collaborative filtering method for top-n recommendation task using biclustering neighborhood approach. The proposed approach explores bicluster similarity in addition to the standard item similarity. A bicluster is a subset of users and web services which forms a dense sub matrix where every user has an interaction with every web service. By exploring the hierarchical relationship between one bicluster with other bicluster, the proposed approach constructs bicluster similarity to acquire local proximity between a recommended service and user. Then this local measure and a global distance measure are combined to create a balanced ranking score for every web service.

In this paper, the background and the related works of web service recommendation using biclustering and user-based clustering are discussed in Section 2. Section 3 discusses the overall system design of biclustering based collaborative filtering algorithm for personalized web service recommendation. Section 4 discusses in detail the proposed approach to collaborative filtering. Section 5 discusses in detail the experimental results. Finally, Section 6 concludes the paper.

2. RELATED WORK

Bin Xu et.al [1]. has used many user-item subgroups each consisting of a subset of items and a group of like-minded users on these items. It was more likely to make preference predictions for a user by means of the correlated subgroups than the entire user-item matrix. The authors also mentioned the problem of Multiclass Co-Clustering (MCoC) and proposed an efficient solution to it. In this paper, a unified framework has been proposed to extend the traditional CF algorithms by utilizing the subgroups information for enhancing the top-N recommendation performance. This approach is an extension of traditional clustering CF models. Experiments on three real world data sets (MovieLens-100K4, MovieLens-1M and Lastfm) are performed and the effectiveness of the proposed approach has been demonstrated. The results showed that using subgroups was a promising way to improve the top-N recommendation performance further for many popular CF methods. Gediminas Adomavicius et. al [2]. has introduced and explored a number of item ranking techniques that can generate recommendations that have considerably higher aggregate diversity across all users while maintaining comparable levels of recommendation accuracy. Experiments are conducted on three datasets namely Movie Lens, Netflix, and Yahoo Movies. In this paper, three techniques are used for rating the prediction, including one model-based (matrix factorization CF) and two heuristic-based (user-based and item-based CF) techniques. The results show that the recommendation techniques provide significant improvement in prediction accuracy with only a few accuracy loss. Breese et al [3]. has presented two model-based algorithms for computing both top-N recommendations and predictions. The first approach follows a probabilistic approach in which the users are clustered and the conditional probability distribution of different items in the cluster is calculated. The chance for the active user belonging to a specific cluster is given in the basket of items is thus expected from the clustering solution and the probability distribution of items in the cluster. The clustering key for this technique is computed using the expectation maximization (EM) principle. The second approach is based on Bayesian network models where each item in the database is modeled as a node having states corresponding to the rating of that item. This paper presents a detailed comparison of two model-based approaches with the user-based approach where the results are based on Bayesian networks model that outperforms the clustering model as well as the user-based scheme. Heckerman et al [4]. has proposed a recommendation algorithm based on dependency networks instead of Bayesian networks. However, the accuracy of dependency networks is lower when compared to Bayesian networks. In this paper, the proposed approach is more efficient to study and have lesser memory requirements. Aggarwal et al [5]. has presented a graph-based recommendation algorithm where the users are denoted as the nodes in a graph and the edges between the nodes denote the degree of similarity between the users. Recommendations for the users are computed by traversing the close by nodes in the graph. The graphical representation of the model is allowed to confine transitive relations that cannot be caught by nearest neighbor techniques. Thus, improved performance is reported in the graphical representation than the user-based schemes. Cantador, I. et al [6]. has proposed a multilayered semantic social network model that offers different views of general interests underlying a group of people. The appropriateness of the proposed model to a collaborative filtering system is analytically studied. Initially, a number of ontology-based user profiles are taken into account with the common

preferences in order to cluster the domain concept space automatically. With the obtained semantic clusters, similarities among individuals are recognized at multiple semantic preference layers. Rashid, A.M. et al [7]. has proposed a technique called ClustKnn which is a simple algorithm that is well suitable for large data sets. The proposed method compresses data tremendously by constructing a simple but efficient clustering model. Recommendations are generated rapidly by using a simple Nearest Neighbor-based approach. The likelihood of ClustKnn is demonstrated both empirically and analytically. ClustKnn provides very good recommender accuracy when compared to other collaborative filtering algorithms apart from being intuitive and highly scalable. George, T.et al [8]. has considered a novel collaborative filtering approach based on a newly proposed weighted co-clustering algorithm that involves parallel clustering of users and items. The paper designs incremental and parallel versions of the co-clustering algorithm and uses it to construct an efficient real-time collaborative filtering framework. The proposed approach is evaluated on large movie and book rating datasets demonstrate that it is possible to attain better accuracy when compared to that of the matrix factorization and correlation based approaches at a much lower computational cost. Xue, G. et al [9]. has presented a novel approach by combining the advantages of model based collaborative filtering and memory based collaborative filtering approaches by introducing a smoothing-based method. This paper presents the clusters generated from the training data provides the basis for neighborhood selection and data smoothing. As an outcome, it provides superior accuracy as well as better efficiency in recommendations. Experimental results on two datasets namely EachMovie and MovieLens show that their new proposed approach constantly outperforms other user based traditional collaborative filtering algorithms. J. Kelleher et al[10]. has presented a collaborative recommender that uses a user-based model to predict user ratings for particular items. The model consists of the summary rating information that is derived from a hierarchical clustering of the users. This paper shows that the coverage is maximal, the accuracy is good and the proposed algorithm is very efficient where the predictions are completed in time that grows individually with the number of ratings and items but only logarithmically with the number of users. B. Sarwar et al[11]. has addressed the performance issues by scaling up the neighborhood arrangement practice through the use of clustering techniques. The sparsity and high cardinality of a collaborative recommender's dataset is the confront to its efficiency. D. Bridge et al [12]. has generalized an existing clustering technique and applied it to a collaborative recommender's dataset to reduce data sparsity and cardinality. Systematic test is completed using several variations that explores the value of partitioning and grouping of the data. Panagiotis Symeonidis et al [13]. has used biclustering techniques to reveal the duality between users and items in both dimensions simultaneously. The paper deals with the nearest biclustering based collaborative filtering algorithm that uses a novel similarity measures in order to attain partly, the matched user's preferences. The paper also focuses on applying nearest biclusters by combining two different types of biclustering techniques namely Bimax and xMotif for coherent and constant biclustering, respectively. The proposed method improves to a large extent where the performance of the CF process is applied on three real-life data sets.

3. SYSTEM IMPLEMENTATION

An overview of biclustering based collaborative filtering algorithm for personalized web service recommendation method (Figure 1)

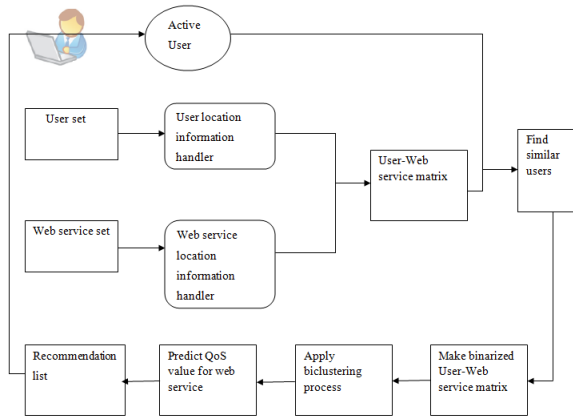


Figure 1: Architecture of the biclustering based collaborative filtering algorithm for personalized web service recommendation

3.1 Module description

Biclustering based collaborative filtering algorithm for personalized Web Service Recommendation method consists of the:

User location information handler: The user location information handler obtains user's location information including the network and the country according to the user's IP address. It also gives support for efficient user-querying based on location.

Service location information handler: The service location information handler gets additional information about the location of the web services according to either their IP addresses or URLs. The location information includes the network and the country in which the Web services are located. It provides functionalities for aiding efficient location-based Web service query.

Find Similar Users: In order to find users who are similar to the active users, consider both the locations as well as users' QoS experiences. For exact user similarity measurement and scalable similar user selection, a weighted user-based PCC is proposed by means of examining QoS variation of Web services and incorporate user locations into similar user selection.

User-based QoS Prediction: The user-based QoS prediction function aggregates the QoS values recognized on target Web services, and predicts the missing QoS values for the active user after a certain number of similar users are discovered for the active user.

Service-based QoS prediction: The service-based QoS prediction function aggregates the QoS values to predict the missing QoS values for the active user after a certain number of similar services are discovered for a target Web service.

Apply biclustering: From the user-web service matrix, a bicluster neighborhood-based approach is applied for making top-n recommendations.

Recommender System: The recommender system function recommends Web services with optimal QoS to the active

user after predicting missing QoS values for all candidate Web services.

4. PROPOSED METHODOLOGY

The Pearson Correlation Coefficient (PCC) has been applied [14] in many recommendation systems to compute the similarity between both users and items. In user-based collaborative filtering, the standard PCC used to measure similarity between two users is computed as:

$$pcc(u, v) = \frac{\sum_{i \in I_v \cap I_u} (r(u, i) - \bar{r}(u))(r(v, i) - \bar{r}(v))}{\sqrt{\sum_{i \in I_v \cap I_u} (r(u, i) - \bar{r}(u))^2} \sqrt{\sum_{i \in I_v \cap I_u} (r(v, i) - \bar{r}(v))^2}} \quad (1)$$

where $I_v \cap I_u$ is the set of web services that are co-invoked by both user v and user u , and $\bar{r}(v)$ and $\bar{r}(u)$ represent the average QoS values that user v and user u have perceived from all the web services they invoked respectively. The similarity value calculated in the above equation is within the continuous range of $[-1, 1]$. The larger the value is, the more similar two users are. However, this equation fails to consider the personal influence of web services on similarity measurement. That is, co-invoked web services are always given equivalent weights in the measurement of similarity between users. Web services with more steady QoS values for all of their users should contribute less to the degree of similarity between users and vice-versa. Therefore, a weighted PCC that incorporates the personality of web services into similarity computation for users has been developed. The following steps are used for computing the weight of Web service i based on its QoS deviation:

QoS normalization. In this step, transformation of each QoS value of web service i , $r(u, i)$, to a real number between 0 and 1 is compared with the maximum and minimum QoS values of i . There are two cases to be considered. If the QoS criterion concerned is positive, i.e., a larger QoS value indicates better QoS, (2) is used to normalize $r(u, i)$; otherwise, if the QoS criterion is negative, i.e., a smaller QoS value indicates better QoS, (3) is used to normalize $r(u, i)$

$$n(u, i) = \frac{r(u, i) - \min r(i)}{\max r(i) - \min r(i)} \quad (2)$$

$$n(u, i) = \frac{\max r(i) - r(u, i)}{\max r(i) - \min r(i)} \quad (3)$$

where $r(i)$ represents the set of QoS values of Web service i . In the case of $\max r(i) = \min r(i)$, where $n(u, i) = 1$ is set.

Standard deviation computation based on normalized QoS values. For a Web service i , its standard QoS deviation after QoS normalization is computed as:

$$d_i = \begin{cases} \sqrt{\sum_{u \in U_i} (n(u, i) - \bar{n}(i))^2 / |U_i|}, & \text{if } |U_i| \geq \theta \\ \sqrt{\sum_{u \in U_i} (n(u, i) - \bar{n}(i))^2 / |U_i| \times \frac{|U_i|}{\theta}}, & \text{if } |U_i| < \theta \end{cases} \quad (4)$$

where $\bar{n}(i)$ is the average QoS value of Web service i , θ is a threshold for the number of users that have invoked i , i.e., U_i . If U_i is very small, the standard deviation is likely to be overestimated by the original standard deviation computation formula. The θ is used to address the above issue.

Weight generation. For a Web service i , its weight is straightforwardly obtained using

$$w_i = d_i \quad (5)$$

The value of w_i is always in the range $[0, 1]$.

After computing the weight of contribution for every Web service, the following formula for computing similarity degree between users' u and v is developed and computed as:

$$\text{sim}(u, v) = \frac{\sum_{i \in I_v \cap I_u} w_i (r(u, i) - \bar{r}(u))(r(v, i) - \bar{r}(v))}{\sqrt{\sum_{i \in I_v \cap I_u} w_i (r(u, i) - \bar{r}(u))^2} \sqrt{\sum_{i \in I_v \cap I_u} w_i (r(v, i) - \bar{r}(v))^2}} \quad (6)$$

The above formula incorporates the personalized influence of Web services into user similarity measurement, and implies that Web services with larger weights will contribute more to the two users' similarity.

As similar as user-based collaborative filtering, previous item-based collaborative filtering methods also often adopt the standard PCC to measure similarity between items. The formula for computing PCC between Web service i and j is computed as:

$$\text{pcc}(i, j) = \frac{\sum_{u \in U_i \cap U_j} (r(u, i) - \bar{r}(i))(r(u, j) - \bar{r}(j))}{\sqrt{\sum_{u \in U_i \cap U_j} (r(u, i) - \bar{r}(i))^2} \sqrt{\sum_{u \in U_i \cap U_j} (r(u, j) - \bar{r}(j))^2}} \quad (7)$$

The working principle for Pearson Correlation Coefficient is computed as:

$$\text{sim}(i, j) = \frac{\sum_{u \in U_i \cap U_j} w_u (r(u, i) - \bar{r}(i))(r(u, j) - \bar{r}(j))}{\sqrt{\sum_{u \in U_i \cap U_j} w_u (r(u, i) - \bar{r}(i))^2} \sqrt{\sum_{u \in U_i \cap U_j} w_u (r(u, j) - \bar{r}(j))^2}} \quad (8)$$

$U_i \cap U_j$ = subset of users that invoked both service i and j

$r(u, i)$ = QoS values of service i

$r(u, j)$ = QoS values of service j

$\bar{r}(i)$ = average QoS values of web service i

$\bar{r}(j)$ = average QoS values of web service j

w_u = weight of user u which is defined to be the standard deviation of the normalized QoS values of u

It takes into account the influence of user personality and computes the similarity between two Web services i and j.

The working principle for predicting user based QoS value is computed as:

$$\hat{r}_u(u, i) = \bar{r}(u) + \frac{\sum_{v \in N(u)} \text{Sim}(u, v) \times (r(v, i) - \bar{r}(v))}{\sum_{v \in N(u)} \text{Sim}(u, v)} \quad (9)$$

The following scenario is considered where an active user is searching for high-quality Web services in a Web service discovery system or the system is recommending high-quality Web services to an active user. In these scenarios, predicting QoS values for Web services unknown to the active user is firstly required; then, Web services with satisfactory QoS can be identified and recommended to the user. This work focuses on predicting QoS values of Web services for recommendation. When an active user is searching for Web services with specified functionality, the predicted QoS values can help the users discover the Web service with optimal QoS value from a set of candidate services. The QoS prediction method can also identify a set of high-quality Web services, and directly recommend them to an active user for selection.

The bicluster neighborhood framework (BCN) for collaborative filtering [15] consists of three basic steps:

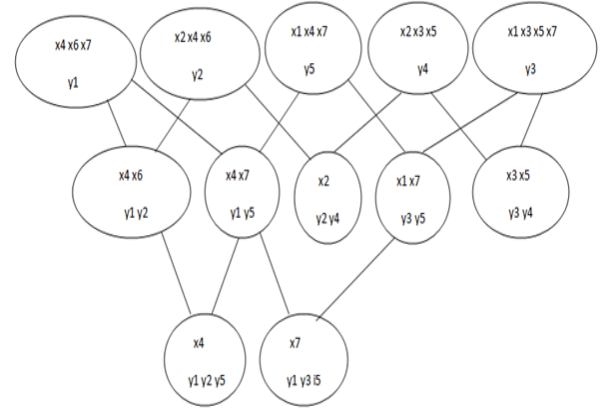
1. Given user x, map x to the smallest bicluster $Z = (X, Y)$ that contains x.
2. Identify candidate set Y of web services for recommendation by exploring the bicluster neighborhood of Y.

3. Rank web services in Y by combining global and bicluster neighborhood similarity.

Consider the following dataset containing 7 users and 5 web services.

Users\Web services	Y1	Y2	Y3	Y4	Y5
x1	0	0	1	0	1
x2	0	1	0	1	0
x3	0	0	1	1	0
x4	1	1	0	0	1
x5	0	0	1	1	0
x6	1	1	0	0	0
x7	1	0	1	0	1

The biclusters corresponding to the above dataset has been represented graphically below.



The algorithm for bicluster neighborhood (BCN) framework for top-n recommendation is given as:

Input: e_i : user to perform recommendation for

Input: K: full data matrix

Input: n: desired num of web service recommendations

Result: Return a size n list of recommended web services

begin

$Z \leftarrow (X, Y) \leftarrow$ Smallest bicluster of e_i $Z_1 \leftarrow$ Lower Neighbors (Z);

$Z_s \leftarrow$ Siblings(z);

candidates $\leftarrow \emptyset$;

for $Z_l \in Z_1$ do

candidates \leftarrow candidates $\cup Y_l/Y$

for $Z_s \in Z_s$ do

candidates \leftarrow candidates $\cup Y_s/Y$

for $y \in$ candidates do

Compute $r(x, y)$

Return top n web services in candidates ranked by r
(x, y)
end

In the final step of the framework, the candidate web services are ranked by combining global and bicluster neighborhood similarity.

Global similarity:

Let (X, Y) and (X', Y') be two biclusters, and web service $y \in Y$ and web service $y' \in Y'$. The global distance between user x and web service y' is defined as:

$$g(x, y') = \frac{\sum_{y \in Y} J(y, y')}{|Y|}$$

where $J(y, y')$ is the Jaccard index, which is defined over all users who interact with y and those who interact with y'. Let X_y be the set of all users who interact with y, and $X_{y'}$ be the set of users for y'.

$$J(y, y') = \frac{|X_y \cap X_{y'}|}{|X_y \cup X_{y'}|}$$

Bicluster similarity:

Let $Z = (X, Y)$ and $Z' = (X', Y')$ be two different biclusters. The union of these two biclusters is defined as $D = (X \cup X', Y \cup Y')$. The zeros-induced similarity measure is

$$b(Z, Z') = 1 - \frac{\text{zeros}(D)}{|D|}$$

where $|D| = |X \cup X'| * |Y \cup Y'|$ and zeros (D) is the number of zeros occurring in the submatrix D. The fewer the zeros the more similar Z and Z' are seemed to be. Assuming Z to be the initial cluster and Z' to be the lower neighbor or sibling of Z, at least one zero must be in D.

Local similarity:

The rank of a candidate web service is computed by aggregating the bicluster similarity of all biclusters in which y occurs to the minimum bicluster. Mathematically it is computed as,

$$l(x, y') = \max_{Z' \in C} b(Z, Z')$$

If recommendation to user u6 has to be made then the smallest bicluster for x6 is ({x4, x6}, {y1, y2}), which has a lower neighbor ({x4}, {y1, y2, y5}). One candidate web service is y5. Then, ranking score for y5 is:

$$\begin{aligned} r(x6, y5) &= g(x6, y5) \times \max_{Z' \in C} b(Z, Z') \\ &= \left(\frac{0.5+0.2}{2}\right) * \left(1 - \frac{2}{8}\right) \\ &= 0.2625 \end{aligned}$$

In order to calculate the ranking score for y1, y2 consider the following data set

Users\Web services	Y1	Y2
x1	0	0
x2	0	1
x3	0	0
x4	1	1
x5	0	0
x6	1	1
x7	1	0

The following approach is used for ranking the web services y1, y2

$$y_n = \frac{\text{Number of non zero entries}}{\text{Total entries}}$$

$$y_1 = \frac{3}{7} = 0.43$$

$$\text{Similarly, } y_2 = \frac{3}{7} = 0.43$$

Final rank = [0.5 * rank of a web service + 0.5 * QoS value of the particular web service]

where equal weight is chosen for the rank of a web service and QoS value of a web service for the final ranking formula.

5. EXPERIMENTAL RESULTS

The web service related data is collected from the URL www.wsdream.com [18]. This dataset contained the QoS records of service invocations on 5825 Web services from 339 users. The dataset can be transformed into a user-service matrix. Each item of the user-service matrix is a pair of values: response time (also called Round Trip Time, RTT) and throughput (TP). Therefore, the original user-service matrix can be decomposed into two simpler matrices: RTT matrix and TP matrix. The RTT matrix or the TP matrix is used to compute both the user and the service similarities. This dataset also contained both the IP addresses of all users and the URLs of all Web services. Experiments were conducted on wsdream dataset in order to predict the accuracy of the proposed system. The standard way of measuring the performance of personalized search is by calculating the precision and recall. 80% of users are used for training while 20% is used for testing.

Precision, Recall and F-measure

The precision is measured by the fraction of the retrieved documents that are relevant to the search,

$$\text{Precision} = \frac{| \{ \text{Relevant documents} \} \cap \{ \text{Retrieved documents} \} |}{| \{ \text{Retrieved documents} \} |}$$

The recall value is measured by the fraction of documents that are relevant to the query and are retrieved successfully,

$$\text{Recall} = \frac{| \{ \text{Relevant documents} \} \cap \{ \text{Retrieved documents} \} |}{| \{ \text{Relevant documents} \} |}$$

F-Measure which combines precision and recall is used to measure the accuracy of the performance of PCC and modified PCC

$$F - \text{Measure} = 2 * \frac{\text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}$$

Table 1. shows the performance comparison of number of recommendations using user-based qos prediction and biclustering where the precision, recall and f-measure gets increased as number of recommendations increases in both the approaches (user-based clustering and biclustering). But, biclustering approach provides better result.

Table 1. Precision, Recall and F-measure table for number of recommendations

Number of recommendations	Precision		Recall		F-measure	
	User-based clustering	biclustering	User-based clustering	biclustering	User-based clustering	biclustering
10	0.65	0.72	0.60	0.83	0.624	0.771
25	0.69	0.75	0.66	0.85	0.674	0.796
50	0.72	0.80	0.72	0.87	0.72	0.833
75	0.75	0.84	0.76	0.90	0.755	0.868

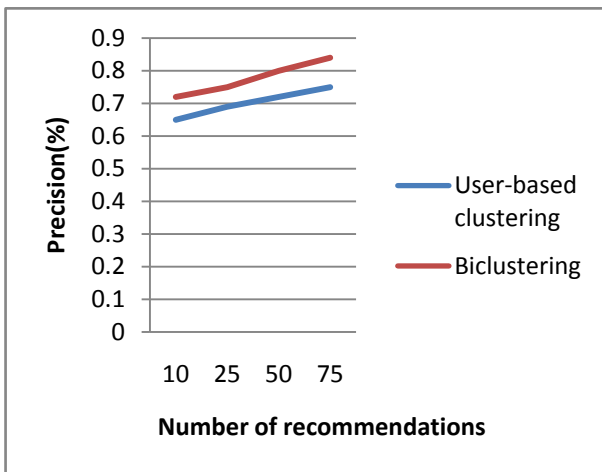


Figure 2. Precision comparison of User-based clustering and biclustering

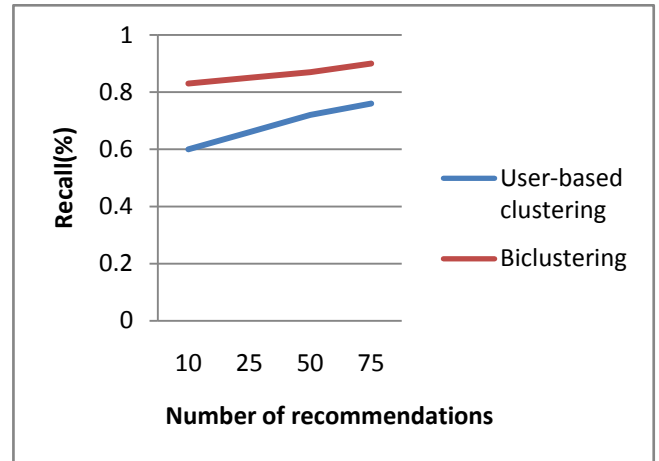


Figure 3. Recall comparison of User-based clustering and biclustering

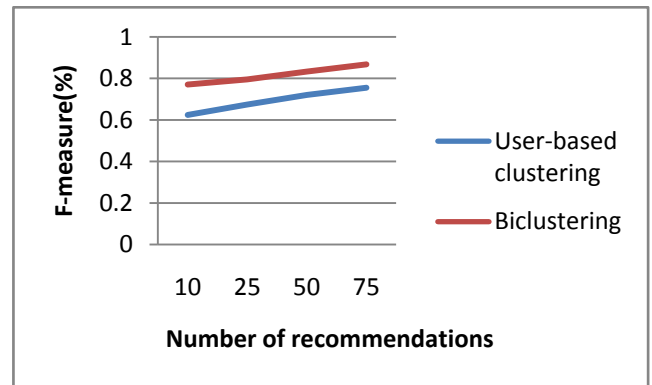


Figure 4. F-measure comparison of User-based clustering and biclustering

The precision and recall values are compared by using two similarity measures namely, user-based QoS prediction and biclustering and it is shown in Figure 2, 3, 4. In this precision-recall graph, biclustering gives better Precision-Recall results.

The sparsity of the input matrix is computed using the number of zero's present in the matrix to the total number of entries. The sparsity is compared with the number of users as shown in Figure.5. Thus, the biclustering approach provides better result.

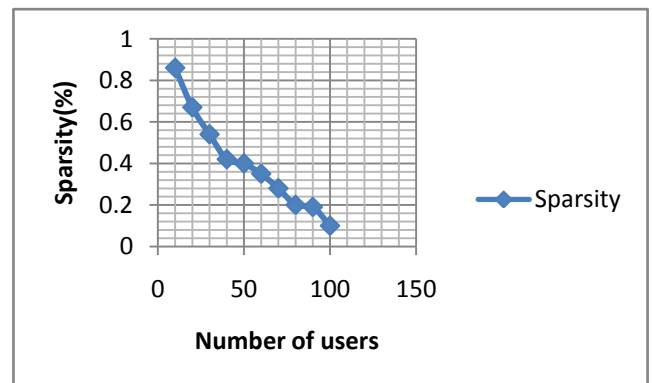


Figure 5. Sparsity for number of users

When CPU time gets increased, the number of users' increases in both the approaches (user-based clustering and biclustering). The CPU time is compared using two

methodologies namely, user-based clustering and biclustering as shown in Figure 6. In this CPU time graph, biclustering gives better results.

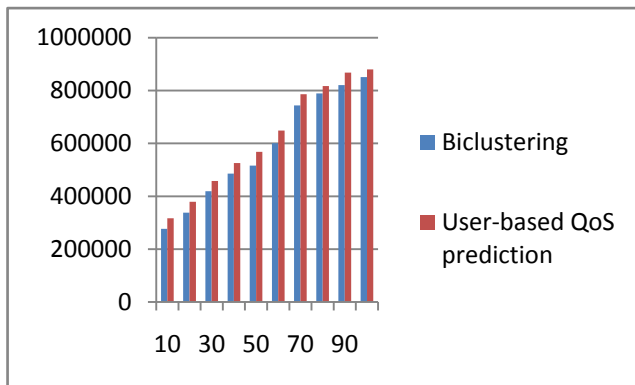


Figure 6. CPU time comparison of User-based QoS prediction and biclustering

6. CONCLUSION

A novel collaborative filtering method based on BCN framework has been proposed and tested for top-n recommendation task. A very accurate recommendation was obtained with Biclustering based collaborative filtering approach when tested with web service data set. When compared with other algorithms, Biclustering based collaborative filtering approach presented better results, since the biclusters give a much higher flexibility on dealing with datasets, as the clustering is not restricted only to users, but involves the attributes as well. With this flexibility it is possible that one user belongs to more than one group at the same time. When compared with other methodologies found in the literature, the proposed approach presented best results. The results confirm that the proposed method can alleviate sparsity problem and improves the recommendation quality.

7. REFERENCES

- [1] Bin Xu Jiajun Bu Chun Chen Deng Cai. 2012. An Exploration of Improving Collaborative Recommender Systems via User-Item Subgroups. International World Wide Web Conference Committee (IW3C2), Lyon, France. ACM 978-1-4503-1229-5/12/04. pp 16–20.
- [2] Gediminas Adomaviciu and YoungOk Kwon. 2012. Improving Aggregate Recommendation Diversity Using Ranking-Based Techniques. IEEE transactions of knowledge data Engineering. Vol.24 (No5).pp.12-23
- [3] Breese, John S., David Heckerman, and Carl Kadie. 1998. "Empirical analysis of predictive algorithms for collaborative filtering." Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc.
- [4] Heckerman, David, et al. 2001. "Dependency networks for inference, collaborative filtering, and data visualization." The Journal of Machine Learning Research1: 49-75.
- [5] Aggarwal, Charu C., et al. 1999. "Horting hatches an egg: A new graph-theoretic approach to collaborative filtering." Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM
- [6] Cantador, Iván, and Pablo Castells. 2006. "Multilayered semantic social network modeling by ontology-based user profiles clustering: application to collaborative filtering." Managing Knowledge in a World of Networks. Springer Berlin Heidelberg, 334-349.
- [7] Al Mamunur Rashid, Shyong K. Lam, George Karypis, and John Riedl. 2006. "ClustKNN: a highly scalable hybrid model-& memory-based CF algorithm." Proceeding of WebKDD
- [8] George, Thomas, and Srujana Merugu. 2005. "A scalable collaborative filtering framework based on co-clustering." Data Mining, Fifth IEEE International Conference on. IEEE,
- [9] Xue, Gui-Rong, et al. 2005. "Scalable collaborative filtering using cluster-based smoothing." Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval. ACM
- [10] Kelleher.J and D. Bridge. 2003. "Rectree: An accurate, scalable collaborative recommender?". In Procs. of the Fourteenth Irish conference on Artificial Intelligence and Cognitive Science, pages: 89–94
- [11] Sarwar, Badrul M., et al. 2002. "Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering." Proceedings of the fifth international conference on computer and information technology. Vol. 1.
- [12] Bridge, Derek, and Jerome Kelleher. 2002. "Experiments in sparsity reduction: Using clustering in collaborative recommenders." Artificial Intelligence and Cognitive Science. Springer Berlin Heidelberg, 144-149.
- [13] Panagiotis Symeonidis, Alexandros Nanopoulos, Apostolos N. Papadopoulos, Yannis Manolopoulos. 2008. "Nearest-biclusters collaborative filtering based on constant and coherent values" information retrieval Volume 11, Issue 1 Pages: 51 - 75
- [14] Jianxun Liu, Mingdong Tang, Member, IEEE, Zibin Zheng, Member, IEEE, Xiaoqing (Frank) Liu, Member, IEEE, Saixia Lyu. 2015. Location-Aware and Personalized Collaborative Filtering for Web Service Recommendation
- [15] Faris Alqadah , Chandan K. Reddy , Junling Hu, Hatim F. Alqadah .Knowl Inf Syst. 2014. Biclustering neighborhood-based collaborative filtering method for top-n recommender systems DOI 10.1007/s10115-014-0771-x
- [16] P. A. D. d. C. de Castro, F. O. d. de Franca, H. M. Ferreira and F. J. V. Zuben, "Applying Biclustering to Perform Collaborative Filtering," Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007), Rio de Janeiro, 2007, pp. 421-426.
- [17] De Castro, Pablo AD, et al. "Evaluating the performance of a biclustering algorithm applied to collaborative filtering-a comparative analysis." Hybrid Intelligent Systems, 2007. HIS 2007. 7th International Conference on. IEEE, 2007.
- [18] www.wsdream.com